Upgrade from Informix 12.10 to Informix 14.10

1) Why you want to upgrade.

- ✓ To get more features
- ✓ To improve performance
- ✓ To secure your data
- ✓ To manage your database through new GUI using Informix HQ
- ✓ To get maximum out of your replication
- ✓ Bug fixes
- ✓ End of support

2) How you can upgrade.

- ✓ Ensure that you meet the operating system and hardware requirements for Informix 14.10.

  (https://www.ibm.com/support/pages/informix-server-system-requirements)

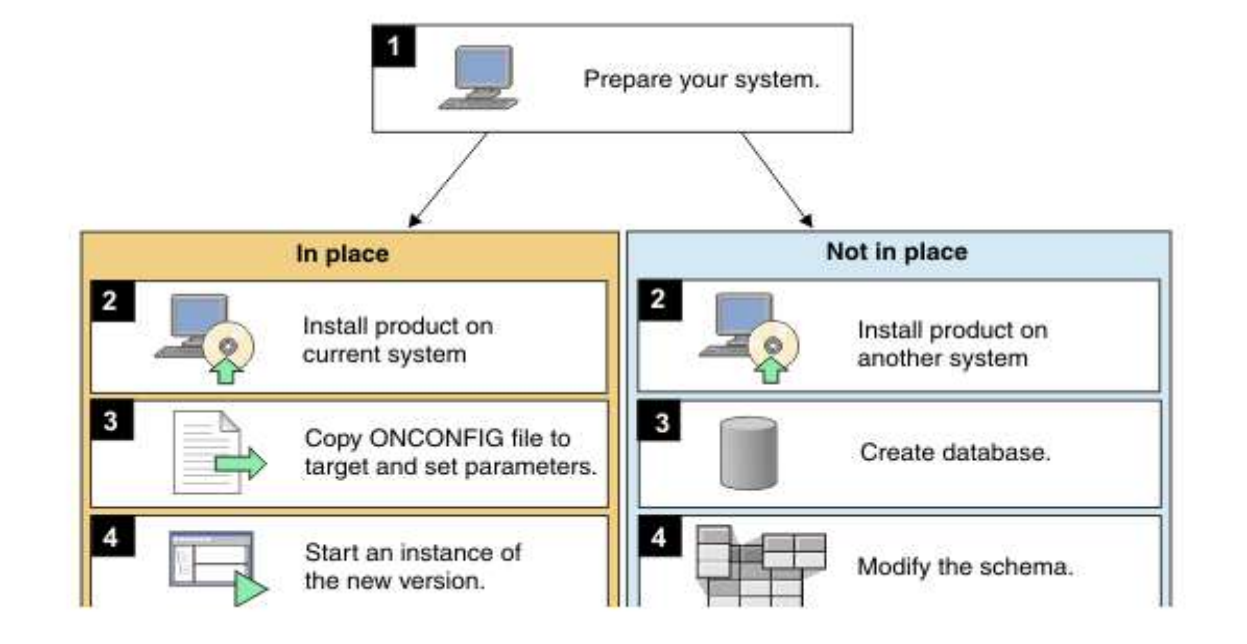
- ✓ There are 2 ways to upgrade/migrate
- ✓ Upgrading (In-place migration)
- ✓ Migrating (Non-in-place migration)

Upgrading v/s Migrating

- **Upgrading (In-place migration)**
  - ✓ uses your existing hardware and operating system
  - ✓ You install the new version in a different location on the same server.
  - ✓ Make changes to $ONCONFIG and $INFORMIXSQLHOSTS
  - ✓ And start the instance, your instance will be automatically converted

- **Migrating (Non-in-place migration)**
  - ✓ The process of “switching over” your environment from one computer to another.
  - ✓ This type of migration requires more planning and setup time compared to upgrading on your existing computer.
  - ✓ Non-in-place migration requires that you modify and copy the database schema, user data, and user objects from one server to another server.
  - ✓ Use this type of migration if you are moving to Informix Version 14.10 from an early version of Informix that has a different architecture, page size, re-architect of dbspaces, and extent allocations

The following illustration shows the differences between in-place and non-in-place migration.



14.10 New Installer

- ✓ Typical installation of IBM Informix product is done by unpacking the downloaded binary and then executing a program named “ids_install”. This program walks you through the licensing terms, asks a couple of questions like which location to install the product and if you want to configure and start up an instance at the end of the install.
- ✓ With IBM Informix 14.10 version things are changing on the way one installs IBM Informix software. A default installation (like you have always done so in the past) will only install the IBM Informix 14.10 Developer Edition. There are additional steps to follow to install the appropriate version needed for your organization’s needs. Let’s see in detail what are the steps that are required to install the correct edition of IBM Informix 14.10
- ✓ One binary with multiple edition (license) installers
- ✓ New InformixHQ software included with the server – $INFORMIXDIR/hq
- ✓ What was removed from the Engine installer (ids_install):

  - JBDC

  - CSDK

  - ESQL
- ✓ Download Informix CSDK from Passport Advantage and install separately.
- ✓ Connection manager is provided within the CSDK

You can download Informix product from IBM website (developer edition for free of cost) and later based on your License type you can download License/edition file from Passport advantage

– Advanced Enterprise Edition (aee)

– Advanced Developer Edition (ade)

– Advanced Enterprise Time Limited Edition (aetl)

– Enterprise Edition (ee)

– Time Limited Edition (tl)

– Workgroup Edition (we)

– Express Edition (e)

– Innovator-C Edition (ie)

– Developer Edition (de)

You can compare these editions from below urls:

https://www.ibm.com/products/informix/editions

https://www.iiug.org/en/2019/10/08/compare-informix/

There are two ways to start the installation:

1) Download 14.10 DE and run ./ids_install. This typically will install Informix Developer edition and later You can change the edition of the installed engine by running the jar file aee_edition.jar
$INFORMIXDIR/jvm/jre/bin/java -jar aee_edition.jar -i console

2) Download 14.10 DE and download the edition.jar file in $INFORMIXDIR and then just run ./ids_install, this will install ids and also invokes applying the license file

First way:

Just run ./ids_install and it will typically install Informix on your server.

```
IBM Informix Dynamic Server Version 14.10.FC4W1DE -- On-Line -- Up 00:00:09 -- 156676 Kbytes

informix@COMP-1081-1:/data1/gaurav/spaces_1410FC4 $ oninit -version
Program Name:           oninit
Build Version:          14.10.FC4W1DE
Build Number:           N001
Build Host:             njdc-lxibm01
Build OS:               Linux 3.10.0-693.el7.x86_64
Build Date:             Tue Jun 16 03:27:59 CDT 2020
Build Timestamp:        2020-06-16T03:09:11-05
GLS Version:            glslib-7.00.FC4
```

Copy aee_edition.jar in new 14.10 $INFORMIXDIR

```
drwxr-xr-x.  3 informix informix      4096 Jul  1 09:46 snmp
-rwxr-xr-x.  1 root     root        197468 Jul  1 09:46 IBM_Informix_14.10_Install_07_01_2020_09_45_54.log
drwxrwxr-x.  4 informix informix      4096 Jul  1 09:46 uninstall
drwxrwxr-x.  3 root     root          4096 Jul  1 09:46 jvm
drwxr-xr-x.  5 informix informix      4096 Jul  1 09:46 bin
drwxrwx---.  2 informix informix      4096 Jul  1 09:46 ssl
drwxr-xr-x.  3 informix informix      4096 Jul  1 09:46 isa
drwxr-xr-x.  2 informix informix      4096 Jul  1 09:46 gskit
drwxr-xr-x.  3 informix informix      4096 Jul  1 09:46 lib
-rwxr-xr-x.  1 root     root         14429 Jul  1 09:46 IBM_Informix_Software_Bundle_Install_07_01_2020_09_45_31.log
-rw-r--r--.  1 informix informix  11804679 Jul  1 10:01 aee_edition.jar
drwxr-xr-x.  2 root     root          4096 Jul  1 10:03 aee_extract
-rwxr-xr-x.  1 root     root          5601 Jul  1 10:03 Informix_Edition_Installer_Install_07_01_2020_10_03_22.log
drwxrwx---.  2 informix informix      4096 Jul  1 10:04 tmp
-rwxr-xr-x.  1 informix informix      5812 Jul  1 10:09 Informix_Edition_Installer_Install_07_01_2020_10_09_40.log
drwxrwxr-x.  4 informix informix      4096 Jul  1 10:11 etc
```

- Run **$INFORMIXDIR/jvm/jre/bin/java -jar aee_edition.jar -i console**
- This will invoke ids_install and will change your version

```
informix@COMP-1081-1:/informix/products/14.10.FC4 $ oninit -version
Program Name:           oninit
Build Version:          14.10.FC4W1AEE
Build Number:           N001
Build Host:             njdc-lxibm01
Build OS:               Linux 3.10.0-693.el7.x86_64
Build Date:             Tue Jun 16 03:27:59 CDT 2020
Build Timestamp:        2020-06-16T03:09:11-05
GLS Version:            glslib-7.00.FC4
```

Preparing for migration

1) Create a new $INFORMIXDIR for 14.10.
2) Create new ONCONFIG based on your ONCONFIG.std and your older 12.10 onconfig and sqlhosts files
3) Run ./ids_install
4) Re-compile any UDRs
5) Check for any pending “in-place alter” on your 12.10. Why?

Why “In-place” alter is so important?

Since version 7.24, some ALTER TABLE statements have been done as "fast alters", where rows are only altered to fit the new schema when the data page they are on is next written to. If there are many alters done, this can lead to the data pages for a table being a variety of versions. When migrating to a later version (or just for the performance benefit) it is a good idea to "complete" the outstanding in-place alters.

- Informix uses in-place alters to speed up database schema changes whenever possible.
- New rows adopt the new schema definition, old rows do not change, therefore conversion is faster
- Resolve in-place alter before an upgrade

The first step is to work out which tables have data pages at old versions:

- Connect to sysmaster database and run the below query:

SELECT ta.dbsname, ta.tabname, pt.partnum, HEX(pt.flags) flag
FROM sysmaster:systabnames ta, sysmaster:sysptnhdr pt
WHERE ta.partnum = pt.partnum AND
BIT_AND(HEX(pt.flags),'00800000'::BINARYVAR) = '00800000';

It will display the list of table names which are pending for in-place alter

```
dbsname  stores_demo
tabname  t1
partnum  1049095
flag     0x00800901

dbsname  stores_demo
tabname  t2
partnum  1049096
flag     0x00800901

dbsname  stores_demo
tabname  t3
partnum  1049097
flag     0x00800901
```

For more information, please refer below URL:

https://www.ibm.com/support/pages/identifying-outstanding-place-alters

How to fix “In-Place” alter

- ✓ First we need to run oncheck -pT <databasename>:<tablename>, it will display the how many pages are on old versions and how many pages are on new version.

- ✓ So let’s take an example of table:

- ✓ I have a table called “migration” which needs an alteration. Right now this table has only 0 versions (which means this table was not altered) and number of pages that version contains.

- ✓ Whenever we alter a table, its version will be incremented

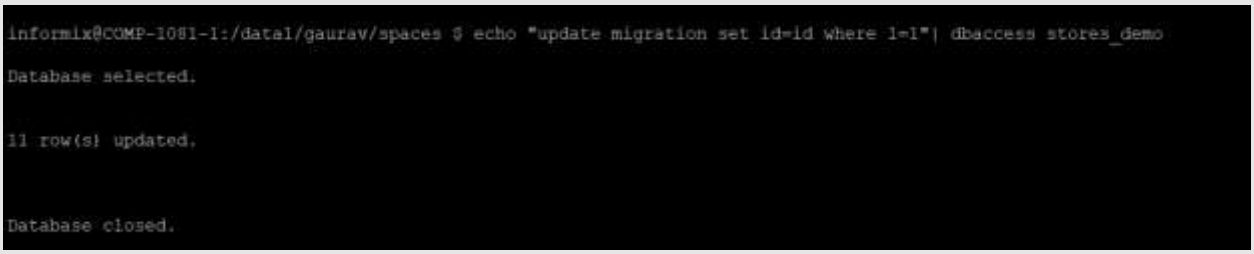
```
TBLspace Usage Report for stores_demo:informix.migration

    Type                 Pages      Empty  Semi-Full       Full  Very-Full
    ---------------- ---------- ---------- ---------- ---------- ----------
    Free                     6
    Bit-Map                  1
    Index                    0
    Data (Home)              1
                     ----------
    Total Pages              8

    Unused Space Summary

        Unused data slots                           242
        Unused bytes per data page                    4
        Total unused bytes in data pages              4

    Home Data Page Version Summary

                 Version                           Count

                       0 (current)                     1
```

- ✓ I will add a new column to this migration table and insert data.
- ✓ Now it will show two versions (0 and 1) . So before upgrading the ids version, it is advisable to fix in-place alter so that both the data pages would be on current version which is 1 in this case.

```
TBLspace Usage Report for stores_demo:informix.migration

    Type                 Pages      Empty  Semi-Full       Full  Very-Full
    ---------------- ---------- ---------- ---------- ---------- ----------
    Free                     5
    Bit-Map                  1
    Index                    0
    Data (Home)              2
    Data (Remainder)         0          0          0          0          0
                     ----------
    Total Pages              8

    Unused Space Summary

        Unused data bytes in Home pages               3946
        Unused data bytes in Remainder pages             0

    Home Data Page Version Summary

                Version                                Count

                      0 (oldest)                           1
                      1 (current)                          1
```

- ✓ Now we run dummy update on the table to bring both the data pages on the current version.

- ✓ Update TABNAME set PKEY=PKEY where 1=1;



So once you ran dummy update, it will show all the data pages would come to current version:

```
TBLspace Usage Report for stores_demo:informix.migration

    Type                 Pages      Empty  Semi-Full       Full  Very-Full
    ---------------- ---------- ---------- ---------- ---------- ----------
    Free                     5
    Bit-Map                  1
    Index                    0
    Data (Home)              2
    Data (Remainder)         0          0          0          0          0
                     ----------
    Total Pages              8

    Unused Space Summary

        Unused data bytes in Home pages               3926
        Unused data bytes in Remainder pages             0

    Home Data Page Version Summary

                Version                              Count

                      0 (oldest)                        0
                      1 (current)                       2
```

6) Verify the integrity of the instances allocated and used pages using the oncheck utility

-Check Reserved Pages

- oncheck –pr

– Check Extents

- oncheck –ce

– Check System Catalog Tables

- oncheck –cc <database name>

– Check Data and Indexes

- oncheck –cDI <Database name>

– Check Smart Large Objects

- oncheck –cs <Sbspace name>
- oncheck –cS <Sbspace name>

7) Perform Whole system Backup

8) Create a Rollback Plan. Just in case if upgrade/migration fails.

What if Upgrade fails!!!

- ✓ By default, the CONVERSION_GUARD configuration parameter is enabled and a temporary directory is specified in the RESTORE_POINT_DIR configuration parameter. These configuration parameters specify information that Informix® can use if an upgrade fails. You can change the default values of these configuration parameters before beginning an upgrade.
- ✓ **Prerequisites**: The directory specified in the RESTORE_POINT_DIR configuration parameter must be empty before the upgrade begins, but not when recovering from a failed update.
- ✓ You can change the value of the CONVERSION_GUARD configuration parameter or the directory for restore point files before beginning an upgrade. The default value for the CONVERSION_GUARD configuration parameter in the ONCONFIG file is (2), and the default directory where the server will store the restore point data is $INFORMIXDIR/tmp. You must change this information before beginning an upgrade. You cannot change it during an upgrade.
- ✓ To change information:
- ✓ If necessary for your environment, change the value of the CONVERSION_GUARD configuration parameter.

- ✓ When the CONVERSION_GUARD configuration parameter is set to 2 (the default value), the server will continue the upgrade even if an error related to capturing restore point data occurs, for example, because the server has insufficient space to store the restore point data.
- ✓ However, if the CONVERSION_GUARD configuration parameter is set to 2 and the upgrade to the new version of the server fails, you can use the onrestorept utility to restore your data.
- ✓ However, if you set the CONVERSION_GUARD configuration parameter to 2, conversion guard operations fail (for example, because the server has insufficient space to store restore point data), and the upgrade to the new version fails, you cannot use the onrestorept utility to restore your data.
- ✓ In the RESTORE_POINT_DIR configuration parameter, specify the complete path name for a directory that will store restore point files.
- ✓ The server will store restore point files in a subdirectory of the specified directory, with the server number as the subdirectory name.
- ✓ If the CONVERSION_GUARD configuration parameter is set to 1 and an upgrade fails, you can run the onrestorept utility to restore the Informix instance back to its original state just before the start of the upgrade.
- ✓ If the CONVERSION_GUARD configuration parameter is set to 1 and conversion guard operations fail (for example, because the server has insufficient space to store restore point data), the upgrade to the new version will also fail.

If any restore point files from a previous upgrade exist, you must remove them before you begin an upgrade.

- ✓ Even if you enable the CONVERSION_GUARD configuration parameter, you should still make level 0 backup of your files in case you need to revert after a successful upgrade or in case a catastrophic error occurs and you cannot revert.
- ✓ The **onrestorept** utility, which you can run to undo changes made during a failed upgrade
- ✓ For more information, please check below url:

  https://www.ibm.com/support/knowledgecenter/SSGU8G_14.1.0/com.ibm.mig.doc/ids_mig_255.htm

Perform Upgrade

- ✓ Once you are done with your pre-checks. You are ready to perform an upgrade.
- ✓ Shutdown all your apps
- ✓ Check there is no connection to the database
- ✓ onmode –l to switch to next logical log and then onmode –c to do a checkpoint.
- ✓ HIGHLY recommended to shutdown using “onmode -uky” to terminate any open transactions that may be present.
- ✓ Make changes in your environment file for $INFORMIXDIR='new_directory_14.10_path'
- ✓ Start your instance and keep an eye on online.log until it says “conversion completed”.

Post upgrade

- ✓ Recommended running your update statistics strategy on all databases.
- ✓ Start your apps and make sure users gets connected to database
- ✓ Take 0 level backup
- ✓ Again verify the integrity of pages using onchecks

14.10 Upgrade – A short Summary

- ✓ Create a new directory for the Informix 14.10.
- ✓ Untar Informix tar file and the License file into the new $INFORMIXDIR
- ✓ Use new $ONCONFIG based off the new onconfig.std and amend it to reflect you previous versions $ONCONFIG settings. And you may copy the old $INFORMIXSQLHOSTS file to the 14.10 $INFORMIXDIR.
- ✓ make changes in your environment file for $INFORMIXDIR='new_directory_14.10_path'
- ✓ And start ids as normal using oninit –vy

✓ Check the online.log and check if reports “converted successfully”

16:10:06 Physical Recovery Started at Page (1:2025).

16:10:06 Physical Recovery Complete: 13 Pages Examined, 13 Pages Restored.

16:10:06 Logical Recovery Started.

16:10:06 10 recovery worker threads will be started.

16:10:06 Going to conv

16:10:07 Physical recovery completed, continuing Conversion

16:10:07 Conv/rev: Started check phase of conversion for component RSAM

16:10:07 *** numVersions=7 targetVersion=30 ***

16:10:07 *** entity_ver=26 targetVersion=30 indexVersion= 0 ***

16:10:07 *** FromVersion=16 ToVersion=18 entity_ver=26 targetVersion=30 indexVersion=0 ***

16:10:07 *** RSAM 1 12.10.xC1 ***

.....

16:10:26 Conversion Completed Successfully

How to upgrade/migrate in cluster environment?

- ✓ If we want to upgrade cluster we can use the same way as we have just done. One server by one server.
- ✓ For migration (it's different from in-place upgrade) you can use below techniques:
  - -Rolling upgrade
  - -cdr migrate server
- ✓ And we have some more very old utilities which take more time.
  - Dbexport/dbimport
  - Load/unload
  - HPL
- ✓ So I will be covering only basic of Rolling upgrade and cdr migrate server

Rolling Upgrade

You can perform a rolling upgrade in a high-availability cluster by temporarily converting the primary and secondary servers to stand-alone Enterprise Replication servers. The upgrade occurs without incurring any downtime because Enterprise Replication supports replication between different versions of the server software. You can use this approach to upgrade to a new major version, or to apply fix packs or interim fixes (PIDs).

- **What is Rolling upgrade ?**
  - ✓ Rolling upgrade is a feature which helps upgrade Informix high availability clusters Online.
  - ✓ Easy to migrate between different OS flavors and different IDS versions
  - ✓ Zero or minimum downtime
  - ✓ Provides a way to transform an HA pair to an ER pair
- **What is the need of Rolling Upgrade**?
  - ✓ Provide a server side infrastructure to support rolling upgrade of HDR or RSS system

- **Main usages of this feature:**
  - ✓ Ability to convert HDR or RSS pair to ER and setup replicate for every table in the system automatically.
  - ✓ Uses ERKEY on tables with no primary key.
  - ✓ One of the nodes continue to be online and allow queries including DML while the other undergo product upgrade

How it Works?

The feature is implemented with 2 new cdr commands:

- **cdr check sec2er [-c primary] [--print] <secondary>**
    - Examines the primary and secondary nodes if split is possible
    - Displays warning and error messages
    - Displays the commands to be executed during split (optional)
- **cdr start sec2er [-c primary] <secondary>**
    - Performs the actual split (using rss2er function in syscdr)
    - Always prints the commands being executed
    - HDR pair will be internally converted to RSS first
    - If successful, 2 stand alone servers will be created replicating via ER
    - For more information please read below URL

https://www.ibm.com/support/knowledgecenter/SSGU8G_14.1.0/com.ibm.mig.doc/ids_mig_271.htm

cdr migrate server

The cdr migrate server command automates data migration task between two or more servers. This command also automates setting up of Enterprise Replication between two Informix server instances.

Data migration options:

- ✓ Static mode-offline data migration
- ✓ Create storage spaces using storage pool
- ✓ Migrate schema and data in parallel
- ✓ Dynamic mode – online data migration
- ✓ Create storage spaces using storage pool
- ✓ Migrate schema and data in parallel
- ✓ Resynchronize data using Enterprise Replication

- **What is cdr migrate server?**
  - Cdr migrate server is a feature which helps upgrade Informix high availability clusters Online.
  - Easy to migrate between different OS flavors and different IDS versions
  - Zero or minimum downtime
  - Provides a way to transform an HA pair to an ER pair

- **What is the need of cdr migrate server?**
  - Automates data migration task between two or more servers across OS and across IDS versions.
  - Automates setting up of ER between two Informix server instances.
  - Create storage spaces using storage pool.
  - Migrate schema and data in parallel.
  - Resynchronize data using Enterprise Replication
  - Zero downtime

How it Works?

- cdr migrate server -s source -t target -p phase [-d database] [--exec]
- This will print commands to stdout i.e which all phases this utility have:

  cdr migrate server -s source -t target --phase all

- To execute commands:

  cdr migrate server -s source -t target --phase all –exec

- https://www.ibm.com/support/knowledgecenter/SSGU8G_14.1.0/com.ibm.erep.doc/ids_erp_cdr_migrate_server.htm

Note: Target server has to be up and running on 14.10 and command will be executed from 14.10 server. My intent was if customer was on 12.10 cluster, they can not do this upgrade in one go (OS upgrade and IDS upgrade or if they are upgrading their hardware) so they can use these technologies to migrate server while their system is up all the time. So on the secondary they can upgrade OS and install Informix 14.10 traditionally and start data migration using these technologies. And if their DBA knows the ER it would be great for them to know which table needs to be sync and these migration techniques are very fancy things.